DIGDATA ML Engineer Code Test

Instructions:

Programming language:
Python, you may use non-standard libraries if needed.

Please time your task 1, 2 and 3 and include the total time in your response.

Task 1

The attached file ***key_metrics.json*** is the json response for a web analytics reporting API. Specifically, it includes key metrics (page views, visits, unique visitors and bounce rate) data for one of our sites.

The structure of the json is irregular. The metric counts for each day can be found in

json["report"]["data"][i]["counts"]

and the metrics definitions are in

json["report"]["metrics"]

We need convert this json into a more useful format:

```
[
        {
                "date": datetime(2016, 11, 13),
                "page_views": 6209372,
                "visits": 2326077,
                // other key metrics
        },

        {
        // other dates to follow
        }
        ...
]
```

Basically a list of dictionaries, with metric definitions as the key, and metric counts as value. Note we also expect the “date” string field to be converted to Python datetime object.

Output: print the final data structure or write to a json file

Task 2

Similarly, the second attachment ***products.json*** includes the products traffic for top countries / cities (for a single day) - in this web analytics API, these dimensions (e.g. product, country and city) are referred to as “elements”, see *json[“report”][“elements”]* . The data for each element is embedded - in this specific case, the data is grouped by “country” first, then each record has “breakdown” for “city”, then breakdown by “product”

we would like to “flatten” the json into the following format:

```
[
        {
                "country": "United States",
                "city": "san francisco (California, United States)",
                "product_name": "AVG AntiVirus Free 2014", "page_views": 215,
                "visits": 77,
        },

        // other country/city/product_name entries to follow, no need to add aggregations

]
```

This way it’s easy to store the data in a database or feed it into statistics / machine learning programs.

Important: Your program should be able to handle any number of dimensions (e.g. when the report response is breakout by Country + City + County + Product + Product Category).

Output: print the final data structure or write to a json file

Task 3

Classification Problem:
Leverage demographic, behavioral and engagement data of in-trial subscribers to determine who is “at risk” of canceling, and not becoming paid subscribers. Given the available tables, provide a detailed description of the approach you will take to do the initial Exploratory Data Analysis, Data Preprocessing, Feature engineering, Model Development and Model Evaluation.

Objective: Use the data available for the first three days, since the user signed up for the 7 day trial, to determine the target variable i.e. the **paid_conversion_score** (How likely the user is to convert to a paid subscription after their trial is complete)

Table 1: Subscription Details
Description: Table stores subscription information of the signup user

Column Name	Description	Example
registration_id	Unique Registration ID	111111
trial_start_dt	Trial Start Date	10/12/2021
signup_plan_cd	Sign Up Plan Code e.g. Limited Commercial, Commercial Free	Commercial Free
signup_device_cd	Device used to sign up: OTT, Mobile, Tablet	OTT
signup_coupon_cd	Coupon used by user (default Null)	XXYYZZ
campaign_cd	Campaign ID which drove the user to signup (default: Null)	BlackFridayDeal
subscription_type	Differentiate between new or returning customer i.e. winback	'NEW' | 'WINBACK'
converted_ind	Did the user convert from trial to paid?	True

Table 2: Video Consumption Details
Description: Table stores video consumption information of the signup user

Column Name	Description	Example
registration_id	Unique Registration ID	111111
device_type_nm	Device used while watching content: OTT, Mobile, Tablet	OTT
video_id	Unique Video/Content ID	55555
video_duration_seconds	Total Duration of the Content in seconds	6000
video_25_pct_ind	User Completed 25% of the content?	True
video_75_pct_ind	User Completed 75% of the content?	False
video_content_complete_ind	Did user watch the complete content?	False
video_content_type_cd	Content Type: Live or DVR	DVR
video_category_nm	Content Category: Sports,Primetime, etc	Sports
video_genre_nm	Content Genre Name	Action
video_show_nm	Content Show Name	Clarice
video_season_nbr	Content Season Number	1
video_episode_nbr	Content Episode Number	5

Table 3: Profile Details
Description: Table stores profile information about the signup user. ViacomCBS lets you create multiple profiles per user.

Column Name	Description	Example
registration_id	Unique Registration ID	111111
profile_nm	Profile Name	User 1

profile_type_cd	Profile Type: Kids, Adults	Adult

Table 4: Demographic Details
Description: Table stores demographic information about the signup user

Column Name	Description	Example
registration_id	Unique Registration ID	111111
birth_year	Birth Year	1990
zipcode	Zip Code	222222

Test Grading Rubric:

1. Meets all specified requirements
2. Is valid runnable code
3. Has appropriate usage of code design patterns and datastructures
4. Has extendable architecture
5. Include a Readme file for the application with run instructions